REMARKS

Request for Continued Examination

Applicant respectfully requests continued examination of the above-indicated application as per 37 CFR 1.114.

Response to Claim Rejections

Rejections of Claims 1 and 7 under U.S.C. 103(a) as being unpatentable over Welfeld in further view of Barlow et al (herein Barlow).

In response to the rejections made on Claims 1 and 7 under U.S.C 103(a) as being unpatentable over Welfeld in further view of Barlow, the applicant has provided the following response.

Applicant asserts that although Welfeld teaches a processor with a data memory, a stack memory and a CPU, and that Barlow teaches a stack point generator producing a software stack pointer to access stack memory, that it is not obvious to combine the two teachings and produce the processor as described in the limitation of claim one of the present invention. This is because when each reference is viewed as a whole, individual elements from the two references are not compatible for combination as otherwise suggested by the Examiner. Further detail in this regard is provided below.

Barlow teaches a "custom microprocessor which has been developed for use in ASIC designs" (Col 3 line 21) "being optimized for integration within an ASIC" (Col 1 line38) having. Fig. 1 shows the basic arrangement with "the core of the processor shown at 100, while the boundary of the integrated circuit is shown at 102" (Col 4 lines30-33). As ASICs are fully integrated application specific processors, they are not modular and components of their architecture cannot be simply extracted and combined with other designs. Fig. 3 of Barlow verifies an integrated low-level design processor by illustrating the various registers and low-level logic components linked together to form the custom ASIC microprocessor, including Y register 308 of processor core 100.

The Examiner has suggested that Barlow discloses a stack point generator further for producing a software stack pointer, and that the stack point generator can be included into the system of Welfeld to obtain the invention as specified in Claim 1 of the present invention, providing Col 15 line 65 – Col 16 line 5 as a reference. According to the Examiner reference, Barlow states " The Y register can be used as a software controlled stack pointer" (Col 15 line 65).

Applicant asserts that because Barlow teaches using a Y register to produce a software stack pointer, that in order to merge the two teachings, the Y register of Barlow must be included into the processor of Welfeld. However, as the Y register is integrated within the processor core 100 of the ASIC design, it would not be possible to simply include the Y register into the processor of Welfeld, as this component is not modular and cannot be extracted from the processor core 100 of Barlow being an integrated ASIC design.

Therefore, applicant asserts that in order to include the Y register (and hence the software stack pointer) of Barlow with the teachings of Welfeld, the processor 130 of Welfeld must be entirely replaced with the processor core 100 of Barlow. However, in doing this, the (hardware) stack pointer functionality (as described in the limitation of Claim 1 "A stack point generator ...for producing stack pointers to access the stack memory" of the present invention) of Welfeld would be lost, as Barlow does not teach an additional (hardware) stack pointer to the Y register software controlled stack pointer. Therefore, applicant asserts that the invention described through the limitation of claim 1 cannot be obtained through combining the teachings of Barlow and Welfeld, as a practical merging of the two teachings would either have a (hardware) stack pointer or software stack pointer, but not both as described through the limitation of claim 1.

Additionally, applicant asserts that there would be no motivation for including the teachings of Barlow into the teachings of Welfeld by replacing the processor 130 of Welfeld with the processor core 100 of Barlow. According to the teachings of Welfeld, the processor 130 is required for "operations comprising a push operation..., a jump operation..., and a pop operation" (Col 3 lines 53-56). A quick inspection of the instruction set for the processor core 100 of Barlow (Col 5 and Col 6) will reveal an absence of PUSH, POP and JUMP commands. Without these commands, applicant asserts that operation of the invention claimed by Welfeld would not operate according to its claimed limitations. Applicant points out that although Barlow teaches a BRA (branch always) command, that it is not equal to a JUMP command, as is well known in the art a Branch is an indication of a short change relative to the current program counter

(within the same memory page), while Jump is an indication of a change in program counter that is not directly related to the current program counter (such as a jump to an absolute memory location or a jump using a dynamic or static table to an alternate memory page), and is free of distance limits from the current program counter.

Furthermore, Welfeld teaches that "fast memory access operations are possible, therefore for eight bit operations" (Col 5 lines 18-19) and "for large numbers of bits... results in increased costs and reduced performance" (Col 5 lines 37-40). This contrasts Barlow which teaches 16 bit operations (See Abstract and Col 4 lines 36-40), where "No interrupts are provided" (See Abstract and Col 4 line 10). Therefore, applicant asserts that including the 16 bit processor core 100 of Barlow with the invention of Welfeld would unnecessarily increase costs and reduce performance, as well as lose the functionality of interrupts for the processing and management of user inputs.

For at least the above-mentioned reasons, applicant asserts that it is not obvious to a person of ordinary skill in the art to include the software controlled stack pointer capability of Barlow in the system of Welfeld to produce the present invention as described in the limitation for Claim 1. Applicant also points out that there would be no motivation to do so, as Barlow does not teach essential processor commands PUSH, POP and JUMP in it's instruction set, as required through the teachings of Welfeld. Also, inclusion of Barlows processor core into the system of Welfeld would unnecessarily increase costs and reduce performance the system of Welfeld, as well as lose the functionality and potential usage of interrupts for user input management. Applicant kindly requests that the Examiner re-evaluate Claim 1 in light of the above reasoning, and make for its allowance. As Claim 7 follows the same rationale as Claim 1, applicant also respectfully requests that the Examiner re-evaluate Claim 7 using the above reasoning and make for its allowance.

Rejections of Claims 2 and 8 under U.S.C. 103(a) as being unpatentable over Welfeld in further view of Barlow and Microsoft Computer Dictionary.

Regarding claim 2, applicant asserts that this claim is dependant upon

independent claim 1. Therefore, should an allowance be made for claim 1, applicant kindly requests that similarly an allowance be made for claim 2, as it is dependant upon claim 1.

Regarding claim 8, applicant asserts that this claim is dependant upon independent claim 7. Therefore, should an allowance be made for claim 7, applicant kindly requests that similarly an allowance also be made for claim 8, as it is dependant upon claim 7.

Rejections of Claims 3 and 9 under U.S.C. 103(a) as being unpatentable over Welfeld in further view of Barlow as applied to Claim 1 above, and in further view of Anderson et al. (US 3,969,724)

Regarding claim 3, applicant asserts that this claim is dependant upon independent claim 1. Therefore, should an allowance be made for claim 1, applicant kindly requests that similarly an allowance be made for claim 3, as it is dependant upon claim 1.

Regarding claim 9, applicant asserts that this claim is dependant upon independent claim 7. Therefore, should an allowance be made for claim 7, applicant kindly requests that similarly an allowance also be made for claim 9, as it is dependant upon claim 7.

Rejections of Claims 4, 5, 10 and 11 under U.S.C. 103(a) as being unpatentable over Welfeld, Barlow and Anderson as applied to Claim 3 above, and in further view of Shima et al. (US 4,332,008)

Regarding claims 4 and 5, applicant asserts that these claims are dependant upon claim 3. Therefore, should an allowance be made for claim 3, applicant kindly requests that similarly allowances be made for claims 4 and 5, as they are both dependant upon claim 3.

Regarding claims 10 and 11, applicant asserts that these claims are dependant upon claim 9. Therefore, should an allowance be made for claim 9, applicant kindly requests that similarly allowances be made for claims 10 and 11, as they are both dependant upon claim 9.

Rejections of Claims 6 and 12 under U.S.C. 103(a) as being unpatentable over Welfeld in further view of Barlow as applied to Claim 1 above, and in further view of Applicant Admitted prior art (AAPA).

Regarding Claims 6 and 12, Examiner has suggested that the APAA discloses wherein the stack pointer generator is further for incrementally increasing the stack pointer to point to a next address when used by the central processing unit, and for decreasing the software stack pointer from a predetermined starting position. Examiner also has suggested it would be obvious to combine the capability of the stack pointer and software stack pointer functioning simultaneously within the stack memory of AAPA and stack memory of Welfeld and Barlow.

Applicant asserts that using the same rationale in the response for Claim 1, that it is not possible to combine the stack pointer of Welfeld and the software stack pointer of Barlow for simultaneous stack operation. The software stack pointer of Barlow operates from Y register embedded into ASIC processor core 100 (Col 15 line 65, Fig. 3). Because an ASIC is an integrated application specific circuit, of which its components and devices cannot be separated, the only possible merger of the two teachings to utilize the software stack pointer is to replace the processor 130 of Welfeld with the processor core 100 of Barlow to include the Y register for software stack operation. However, since Barlow does not teach of a (hardware) stack pointer, or any additional stack pointers, only the software stack pointer can exist in a practically combined system of both teachings, rendering simultaneous hardware and software stack operation impossible.

For at least the above-mentioned reason, applicant asserts that it is not obvious to combine the capability of the stack pointer and software stack pointer functioning simultaneously within the stack memory of AAPA and stack memory of Welfeld and

Barlow. Additionally, as Claims 6 and 12 are dependant upon Claims 1 and 7, should allowances be made on Claims 1 and 7, applicant respectfully requests that the Examiner re-evaluate Claims 6 and 12 for their allowances as well.

5 **New Claims**

Claim 13 was added to claim an additional embodiment of the present invention. No new subject matter was introduced in this claim, and all limitations are fully supported in the original disclosure.

10

Sincerely yours,

Date: 05/11/2006

15 Winston Hsu, Patent Agent No. 41,526
P.O. BOX 506, Merrifield, VA 22116, U.S.A.
Voice Mail: 302-729-1562
Facsimile: 806-498-6673
e-mail : winstonhsu@naipo.com

20

Note: Please leave a message in my voice mail if you need to talk to me. (The time in D.C. is 12 hours behind the Taiwan time, i.e. 9 AM in D.C. = 9 PM in Taiwan.)